Herding cats: Program management in communities

Ben Cotton

DevConf.US

Let’s talk about it

- Nice things: @FunnelFiasco
- Not-nice things: `/dev/null`

What is program management?

But first:
What is *project* management?

Dictionary time

Project

- Single, focused result
- Has a defined end
- Output-focused

Program

- Made of projects
- No defined end
- Outcome-focused

A note on terms

What is project management?

“The application of knowledge, skills, tools, and techniques to project activities to meet the project requirements”

— Project Management Institute

What is project management?

- Working with project teams to balance constraints:
  - Time
  - Cost
  - Scope
  - Quality

“Everyone does project management, some just do it poorly.”

— Ben Cotton

What is program management?

What is program management?

“The application of knowledge, skills, tools, and techniques to meet program requirements”

— Project Management Institute

What is program management?

- Like project management, but more of it
- More coordination between projects/activities
- Less involvement in the specific details

EXCEPT!

“bcotton is a default volunteer in Fedora”

— anonymous

What is program management?

Time

Managing Schedules

- Managing schedules doesn't necessarily mean *setting* schedules
- Managing schedules does not mean being held responsible for execution

Managing Schedules

- Managing schedules is
  - Building the schedule
  - Communicating the schedule
  - Updating the schedule
  - Consulting on schedule-related decisions

Why have a schedule?

- Users care
- Downstreams care
- Upstreams probably don't care, but might
- How else will you know it's done?

Types of schedules

- Calendar-based
- Feature-based
- Meh, it's done I guess

Calendar-based schedules

- Start with your target release date and work backward
- How do you pick a date?
  - Upstream or downstream release dates
  - Tradition
  - Conferences/events
  - Fun dates (e.g. Pi day)
  - The amount of time it takes for the new release to be different enough

Feature-based schedules

- Start with your target feature set and work forward
- How do you pick your feature set?
  - How different do you want it to be?
    - What "justifies" a new release?
    - How much change is too much?
  - How interdependent are your changes?
  - How long do you want it to take?
    - YOU CAN'T AVOID THE CALENDAR

Meh, it’s done I guess schedules

- You’re on your own here

Common considerations

- Milestones
  - Feature proposal/code complete deadlines
  - Merge windows
  - Testing
  - Releases (alpha/beta/GA)
- Conflicts
  - Conferences
  - Holidays

Common considerations

- Schedule changes
  - If you move one date, you impact others
- Public perception
  - Not all one week slips are created equal
  - Marketing is a part of the schedule, whether you like it or not

When your schedule is wrong

- Calendar based
  - Cut troublesome features
  - Slip the release date
- Feature based
  - Cut troublesome features
- Meh, it's done I guess
  - The schedule is always right!

When your schedule is wrong

- But what if I was too pessimistic
  - Stop lying, you’re never too pessimistic
  - Release early
  - Add more stuff
  - Relax

Cost

Cost is people

- Community projects don’t necessarily have dollar costs
- ...but people have time costs
- ...but you don’t have control over the people
- ...so your job is helping the people coordinate

Communication is key

- There’s so much information out there
- Distilling it into highlights helps

Meetings meetings meetings

- Meetings can be good
  - Text, phone, or video
  - Take notes (text-based meetings allow bots to take notes for you!)
  - Have an agenda and stick to it
- Don’t let meetings be the only place decisions are made
  - Not everyone can attend your meeting!

Channel flipping

- Pick one synchronous tool and one asynchronous tool
- Keep the barrier to entry low
- Archives are (usually) your friend
- Moderate a channel for important messages

Scope

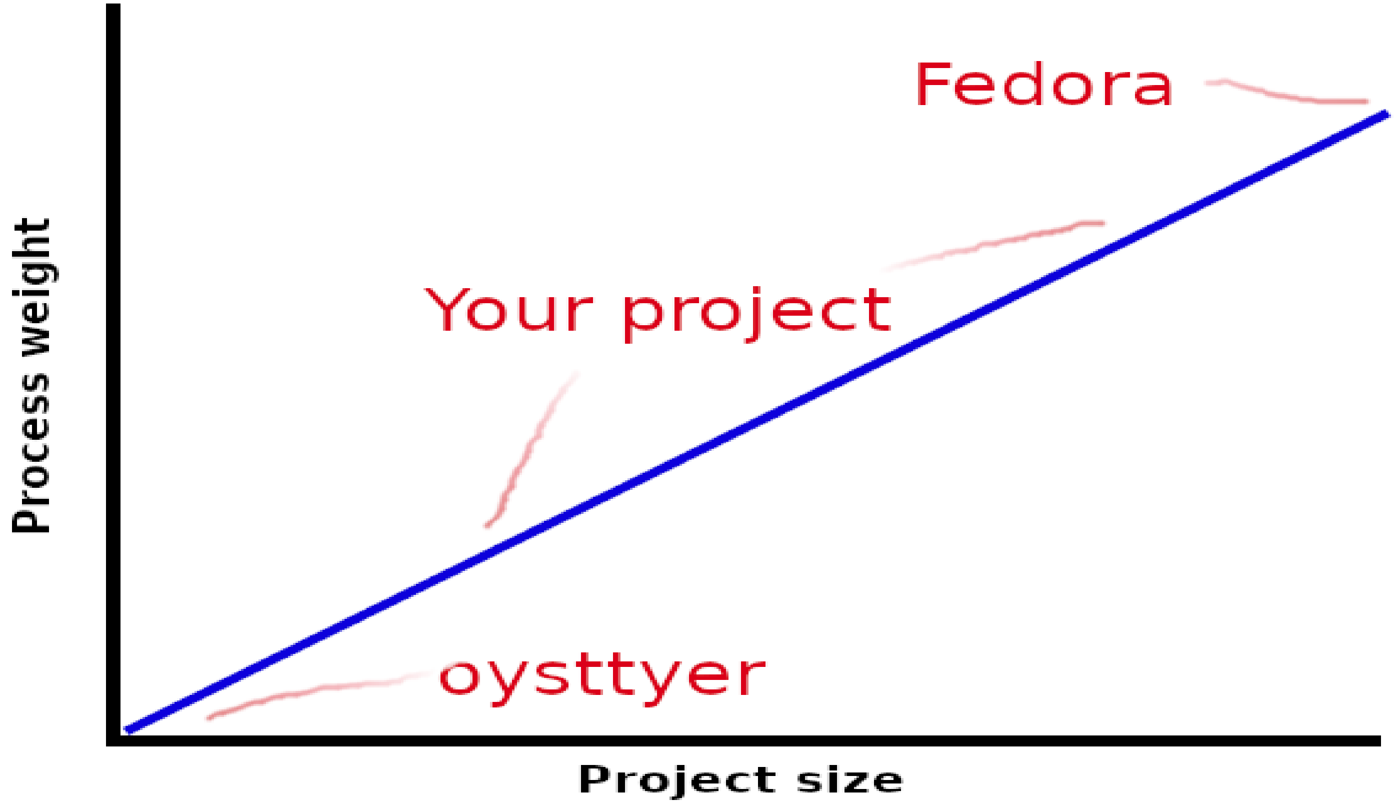
Why have a change process?

- Communication
- Feedback
- More communication

Process varies by size

- Size ≈ number of contributors
- “Weight” of process is proportional to size
- Number of communication channels is exponentially related to number of people

Process varies by size



Considerations

- Who should validate/vet the change?
  - Release engineering?
  - Legal?
  - The community?
- Who approves the change?
  - Community vote?
  - Technical steering body?
  - Project leader?

Opinion time!

- Democracy is messy
- Establish an elected technical body to approve changes (if your project is big enough to justify it)

Considerations

- Conflicting changes
- Broken changes
- Undelivered changes

Quality

Quality

- Tests are good
- Have release criteria
- Triage important-but-not-blocker bugs

How?

How?

- Communicate
- Communicate
- Communicate

How?

- Sit in on meetings and read mailing lists
- Visibly communicate to the community
  - Blog posts
  - IRC office hours
  - Public issue trackers/Kanban boards/etc

How is it different in communities?

Like in companies

- People don’t like process and bureaucracy
- You might not have direct authority
- The job is all about communication and coordination

But in communities

- You can only lead by influence
- It takes more time to build credibility
- You have to show the community your value

The process is here to serve the community; the community is not here to serve the process.

Let’s talk about it

- Nice things: @FunnelFiasco
- Not-nice things: `/dev/null`

Herding cats:
Program management in communities

Ben Cotton

DevConf.US

Let’s talk about it

- Nice things: @FunnelFiasco
- Not-nice things: `/dev/null`

What is program management?

But first:
What is *project* management?

Dictionary time

Project	Program
• Single, focused result	• Made of projects
• Has a defined end	• No defined end
• Output-focused	• Outcome-focused

A note on terms

What is project management?

“The application of knowledge, skills, tools, and techniques to project activities to meet the project requirements”

— Project Management Institute

What is project management?

- Working with project teams to balance constraints:
  - Time
  - Cost
  - Scope
  - Quality

“Everyone does project management, some just do it poorly.”

— Ben Cotton

What is program management?

What is program management?

“The application of knowledge, skills, tools, and techniques to meet program requirements”

— Project Management Institute

What is program management?

- Like project management, but more of it
- More coordination between projects/activities
- Less involvement in the specific details

EXCEPT!

“bcotton is a default volunteer in Fedora”

— anonymous

What is program management?

Time

Managing Schedules

- Managing schedules doesn’t necessarily mean *setting* schedules
- Managing schedules does not mean being held responsible for execution

Managing Schedules

- Managing schedules is
  - Building the schedule
  - Communicating the schedule
  - Updating the schedule
  - Consulting on schedule-related decisions

Why have a schedule?

- Users care
- Downstreams care
- Upstreams probably don't care, but might
- How else will you know it's done?

Types of schedules

- Calendar-based
- Feature-based
- Meh, it's done I guess

Calendar-based schedules

- Start with your target release date and work backward
- How do you pick a date?
  - Upstream or downstream release dates
  - Tradition
  - Conferences/events
  - Fun dates (e.g. Pi day)
  - The amount of time it takes for the new release to be different enough

Feature-based schedules

- Start with your target feature set and work forward
- How do you pick your feature set?
  - How different do you want it to be?
    - What "justifies" a new release?
    - How much change is too much?
  - How interdependent are your changes?
  - How long do you want it to take?
    - YOU CAN'T AVOID THE CALENDAR

Meh, it's done I guess schedules

- You're on your own here

Common considerations

- Milestones
  - Feature proposal/code complete deadlines
  - Merge windows
  - Testing
  - Releases (alpha/beta/GA)
- Conflicts
  - Conferences
  - Holidays

Common considerations

- Schedule changes
  - If you move one date, you impact others
- Public perception
  - Not all one week slips are created equal
  - Marketing is a part of the schedule, whether you like it or not

When your schedule is wrong

- Calendar based
  - Cut troublesome features
  - Slip the release date
- Feature based
  - Cut troublesome features
- Meh, it's done I guess
  - The schedule is always right!

When your schedule is wrong

- But what if I was too pessimistic
  - Stop lying, you’re never too pessimistic
  - Release early
  - Add more stuff
  - Relax

Cost

Cost is people

- Community projects don’t necessarily have dollar costs
- ...but people have time costs
- ...but you don’t have control over the people
- ...so your job is helping the people coordinate

Communication is key

- There’s so much information out there
- Distilling it into highlights helps

@FunnelFiasco

Meetings meetings meetings

- Meetings can be good
  - Text, phone, or video
  - Take notes (text-based meetings allow bots to take notes for you!)
  - Have an agenda and stick to it
- Don’t let meetings be the only place decisions are made
  - Not everyone can attend your meeting!

Channel flipping

- Pick one synchronous tool and one asynchronous tool
- Keep the barrier to entry low
- Archives are (usually) your friend
- Moderate a channel for important messages

Scope

Why have a change process?

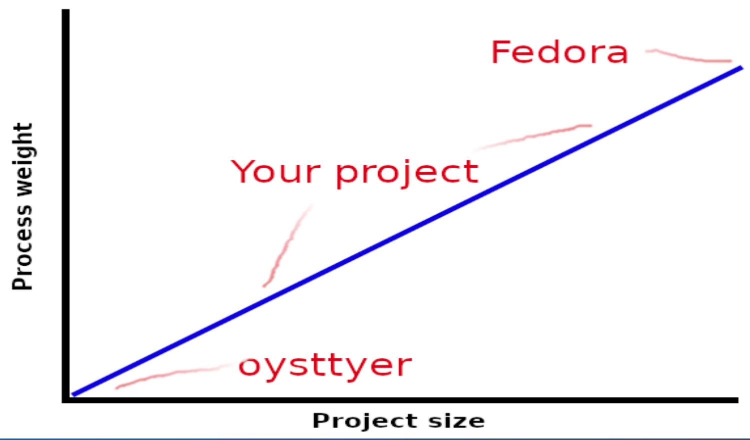
- Communication
- Feedback
- More communication

@FunnelFiasco

Process varies by size

- Size ≈ number of contributors
- “Weight” of process is proportional to size
- Number of communication channels is exponentially related to number of people

Process varies by size



@FunnelFiasco

Considerations

- Who should validate/vet the change?
  - Release engineering?
  - Legal?
  - The community?
- Who approves the change?
  - Community vote?
  - Technical steering body?
  - Project leader?

Opinion time!

- Democracy is messy
- Establish an elected technical body to approve changes (if your project is big enough to justify it)

Considerations

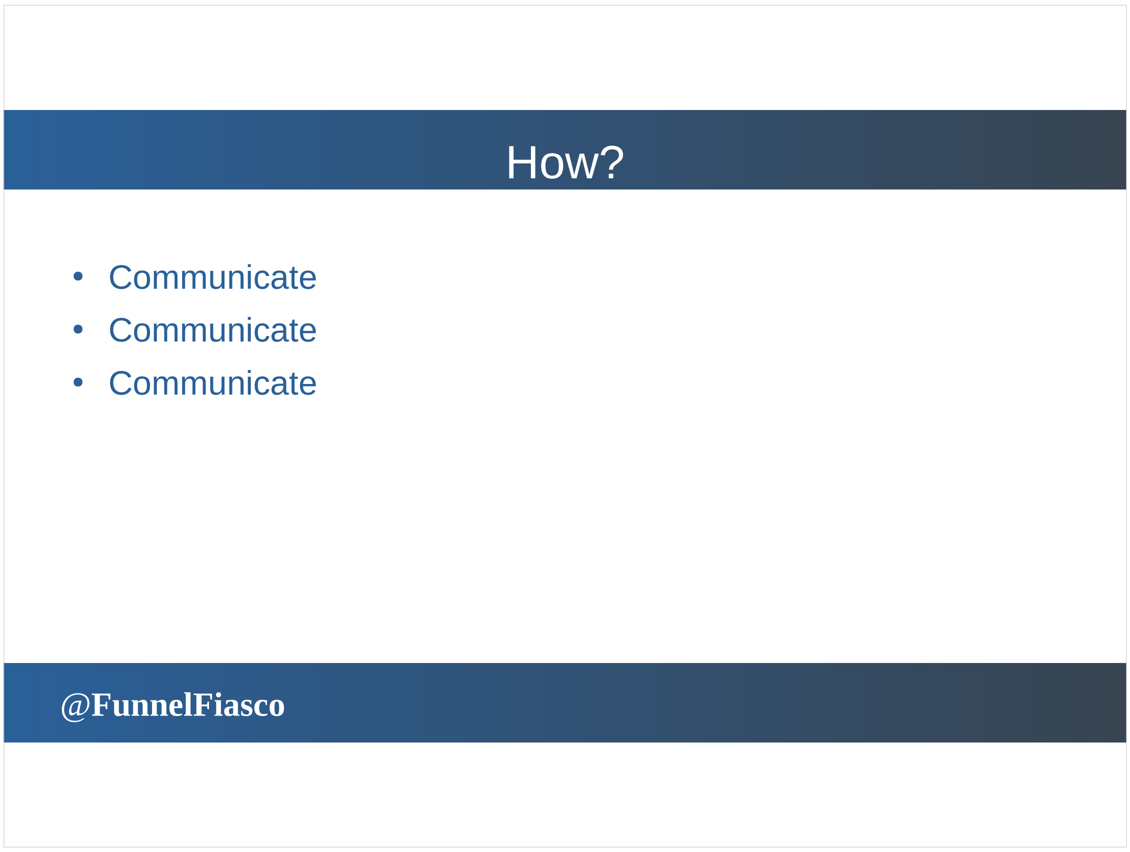
- Conflicting changes
- Broken changes
- Undelivered changes

Quality

Quality

- Tests are good
- Have release criteria
- Triage important-but-not-blocker bugs

How?



How?

- Communicate
- Communicate
- Communicate

@FunnelFiasco

How?

- Sit in on meetings and read mailing lists
- Visibly communicate to the community
  - Blog posts
  - IRC office hours
  - Public issue trackers/Kanban boards/etc

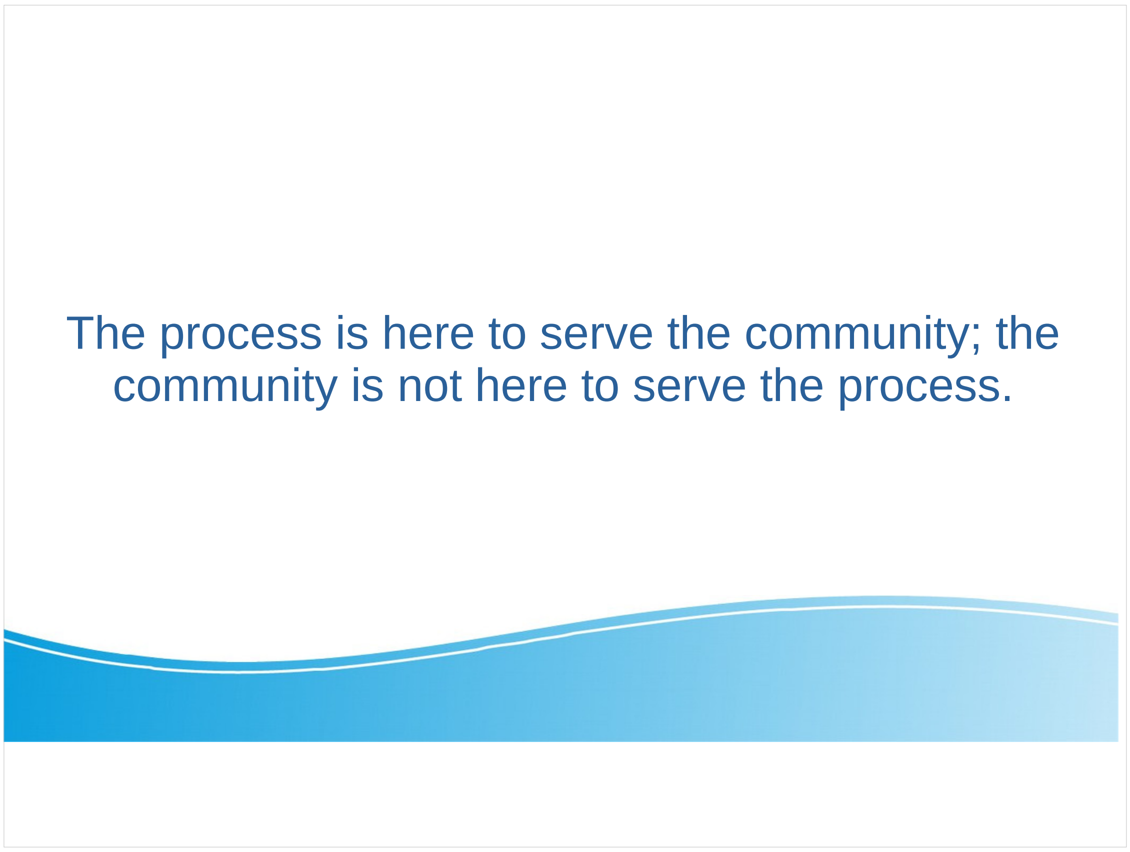
How is it different in communities?

Like in companies

- People don’t like process and bureaucracy
- You might not have direct authority
- The job is all about communication and coordination

But in communities

- You can only lead by influence
- It takes more time to build credibility
- You have to show the community your value



The process is here to serve the community; the community is not here to serve the process.

Let’s talk about it

- Nice things: @FunnelFiasco
- Not-nice things: `/dev/null`